“ExerLink - Enabling Pervasive Social Exergames with Heterogeneous Exercise Devices”

PRESENTED BY SALIKH BAGAVEYEV

Introduction

Authors:
Taiwoo Park, Inseok Hwang, Uichin Lee†, Sunghoon Ivan Lee‡, Chungkuk Yoo, Youngki Lee, Hyukjae Jang, Sungwon Peter Choe, Souneil Park and Junehwa Song

Department of Computer Science, KAIST, †Department of Knowledge Service Engineering, KAIST and ‡Computer Science Department, UCLA
Introduction

• Diverse pervasive devices as game interfaces. Exergames.
• Social exergames provide ubiquitous social interactions
• Focus on Repetetive-Individual-Aerobic exercises
Motivation

• Complement existing persuasive techniques relying on peer pressure and social recognition
• Existing approaches cannot be fully extended to pervasive social exergame scenarios
• Motivating Scenario: three friends prefer various exercises and cannot meet regularly
• Repetetive-Individual-Aerobic exercise benefits:
  Popularity
  Ease of access
  Monotony
  Long-lasting
  Measurability
• Target Social Exergames
Exerlink Platform
Exerlink Platform

- Game Input Controller
- Voice Channel Manager
- Network Manager
- Exercise Information Manager
Providing Fair Gameplay: converting intensity into game input

*Personalized Mapping*
Providing Fair Gameplay: converting intensity into game input

Mapping functions
Providing Fair Gameplay: balancing heterogeneity

- Preferred range for balancing heterogeneity
- Create target rate for acceleration and deceleration
- Adjust acceleration and deceleration to target rate

```
Function: AccelerationBalancing
Input: Vin is the original game input, t is current time, β is the target rate
Output: Vout is the converted game input

1. // detect acceleration
2. if ((Vin - Vin_old) / (t - t_old) > acceleration_threshold)
3.    if (current_state != ACCELERATING)
4.       current_state ← ACCELERATING
5.       t_accel_begin ← t_old
6.      Vin_accel_begin ← Vin_old
7.    // convert acceleration rate from the beginning of the current acceleration
8.      current_accel ← (Vin - Vin_accel_begin) / (t - t_accel_begin)
9.      converted_accel ← β * current_accel
10. else
11.    if (current_state == ACCELERATING)
12.      current_state ← ACCEL_END_WAIT
13.    if (current_state == ACCELERATING || current_state == ACCEL_END_WAIT)
14.      Vout ← Vout_old + converted_accel × (t - t_old)
15.      Vin_old ← Vin
16.      Vout_old ← Vout
17.    t_old ← t
```
Providing Fair Gameplay: balancing delay variation

• Dead reckoning to predict the state of another player

• Fair-ordering service: at the server side action messages should be properly ordered based on players’ reaction times
Controller prototype: hula hoop
Controller prototype: hula hoop
Controller prototype: jump rope
Controller prototype: treadmill and stationary bike
Evaluation

- Point select task (Fitts’ law tests)
- High and low intensity targets
- Collect movement times and error
Results

*Effects:*

Movement time and learning
Results

Effects:

Throughput and SD of errors
Results

Effects:

Movement direction
Results

Effects:

Gender
Results

Case Study 1: Balancing and Fairness

<table>
<thead>
<tr>
<th></th>
<th>Homogeneous Team</th>
<th>Heterogeneous Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Playing time (sec)</td>
<td>86.30</td>
<td>86.35</td>
</tr>
<tr>
<td># of acquired items</td>
<td>3.35</td>
<td>3.40</td>
</tr>
<tr>
<td># of obstacle collisions</td>
<td>3.30</td>
<td>3.30</td>
</tr>
<tr>
<td># of wins</td>
<td>11</td>
<td>9</td>
</tr>
</tbody>
</table>
Results

Case Study 2: Social Interactions

![Bar chart showing the number of interactions for co-located play and remote play.](chart.png)
Conclusion

- Disparate exercises can be turned into fun collaborative activities
- Fair game experiences are achievable
- Players with different exercises can effectively collaborate and compete well while playing the same game.
- Remote exergames did not hinder gaming process, yet promoted more engagement in conversation and exercise.